PBL PROJECT

On

IOT ENERGY METER

Submitted to JNTU HYDERABAD

In Partial Fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted
By

U Keerthana	**(208R1A05H6)**
Y Rushitha	**(208R1A05H8)**
K Akshaya	**(208R1A05E8)**
MD Abdul adil	**(208R1A05GO)**

Under the Esteemed guidance of

Mrs. M.Prashanti

Assistant Professor, Department of CSE

Department of Computer Science & Engineering
CMR ENGINEERING COLLEGE
(*Accredited by NBA,* Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 40

2022-2023

CMR ENGINEERING COLLEGE

(Accredited by NBA, Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, R.R Dist, Hyderabad-501 401)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **“IOT ENERGY METER”**is a bonafide work carried out by

U Keerthana	**(208R1A05H6)**
Y Rushitha	**(208R1A05H8)**
K Akshaya	**(208R1A05E8)**
MD Abdul adil	**(208R1A05G0)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide

Mrs. M.Prashanti
Assistant Professor
Department of CSE
CMREC, Hyderabad

Head of the Department

Dr. Sheo Kumar
Professor & H.O.D
Department of CSE
CMREC, Hyderabad

DECLARATION

This is to certify that the work reported in the present project entitled **"IOT ENERGY METER "** is a record of bonafide work done by us in the Department of Computer Science and Engineering, CMR Engineering College,JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

U. Keerthana	**(208R1A05H6)**
Y. Rushitha	**(208R1A05H8)**
K. Akshaya	**(208R1A05E8)**
MD Abdul Adil	**(208R1A5G0)**

ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. Sheo Kumar**, HOD, **Department of CSE, CMR Engineering College** for their constant support**.**

We are extremely thankful to **Mrs. M.Prashanti ,** Assistant Professor, Internal Guide, Department of CSE, for her constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, we are very much thankful to my parents who guided me for every step.

U. Keerthana	**(208R1A05H6)**
Y. Rushitha	**(208R1A05H8)**
K. Akshaya	**(208R1A05E8)**
MD Abdul Adil	**(208R1A5G0)**

ABSTRACT

The project aims at developing a system which helps in monitoring the readings from an energy meter and controlling the switching of energy meter. This system also has tamper switch, which helps in illegal removing of energy meter cabinet and alerts the authorities in the form of text message. This also sends data to webpage in real-time with tamper alert status too.The controlling device is a Microcontroller. IOT modem, Relay, LCD, tamper switch and energy meter are interfaced to Microcontroller. The microcontroller is programmed such that it sends the energy readings to the authorities by sending simple SMS to the system. It helps controlling the energy meter along with tampering proof facility. The readings are displayed on LCD. The Microcontroller is loaded with intelligent program written using Embedded 'C' language.The modules in the project are: IOT modem for establishing communication between system at house and electricity department, Energy meter which continuously gives usage details, LCD to display current reading of meter, Relay to disconnect the power in case of nonpayment of bill.

CHAPTER 1
INTRODUCTION

1.1 Introduction:

Monitoring and keeping tracking of your electricity consumption for verification is a tedious task today since you need to go to meter reading room and take down readings. Well it is important to know if you are charged accordingly so the need is quite certain. Well we automate the system by allowing users to monitor energy meter readings over the internet. Our proposed system uses energy meter with microcontroller system to monitor energy usage using a meter. The meter is used to monitor units consumed and transmit the units as well as cost charged over the internet using wifi connection. This allows user to easily check the energy usage along with the cost charged online using a simple web application. Thus the energy meter monitoring system allows user to effectively monitor electricity meter readings and check the billing online with ease.

An embedded system is a combination of software and hardware to perform a dedicated task. Some of the main devices used in embedded products are Microprocessors and Microcontrollers.

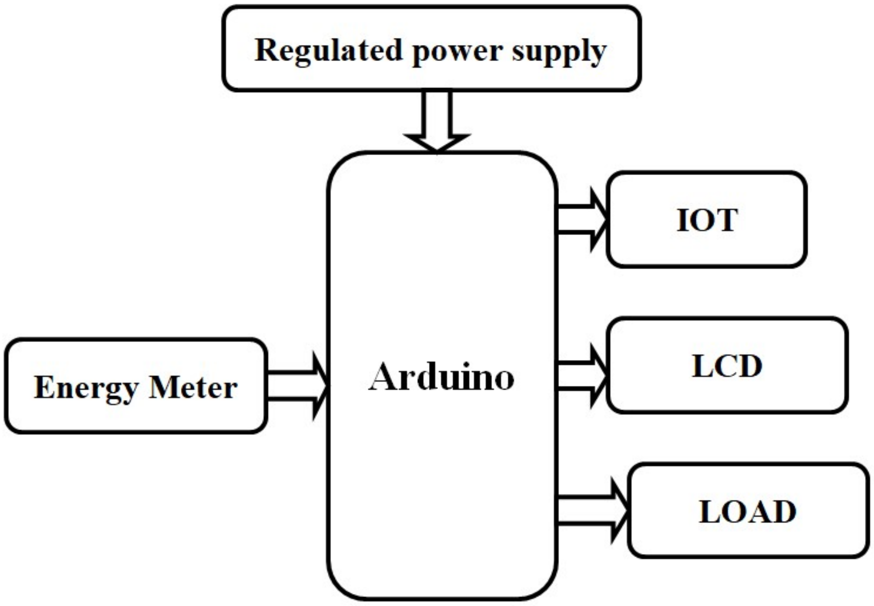
Microprocessors are commonly referred to as general purpose processors as they simply accept the inputs, process it and give the output. In contrast, a microcontroller not only accepts the data as inputs but also manipulates it, interfaces the data with various devices, controls the data and thus finally gives the result.

The “**IOT ENERGY METER MONITORING**” using arduino microcontroller is an exclusive project which is used to designing a completely automated for physically disabled persons.

CHAPTER 2: LITERATURE SURVEY

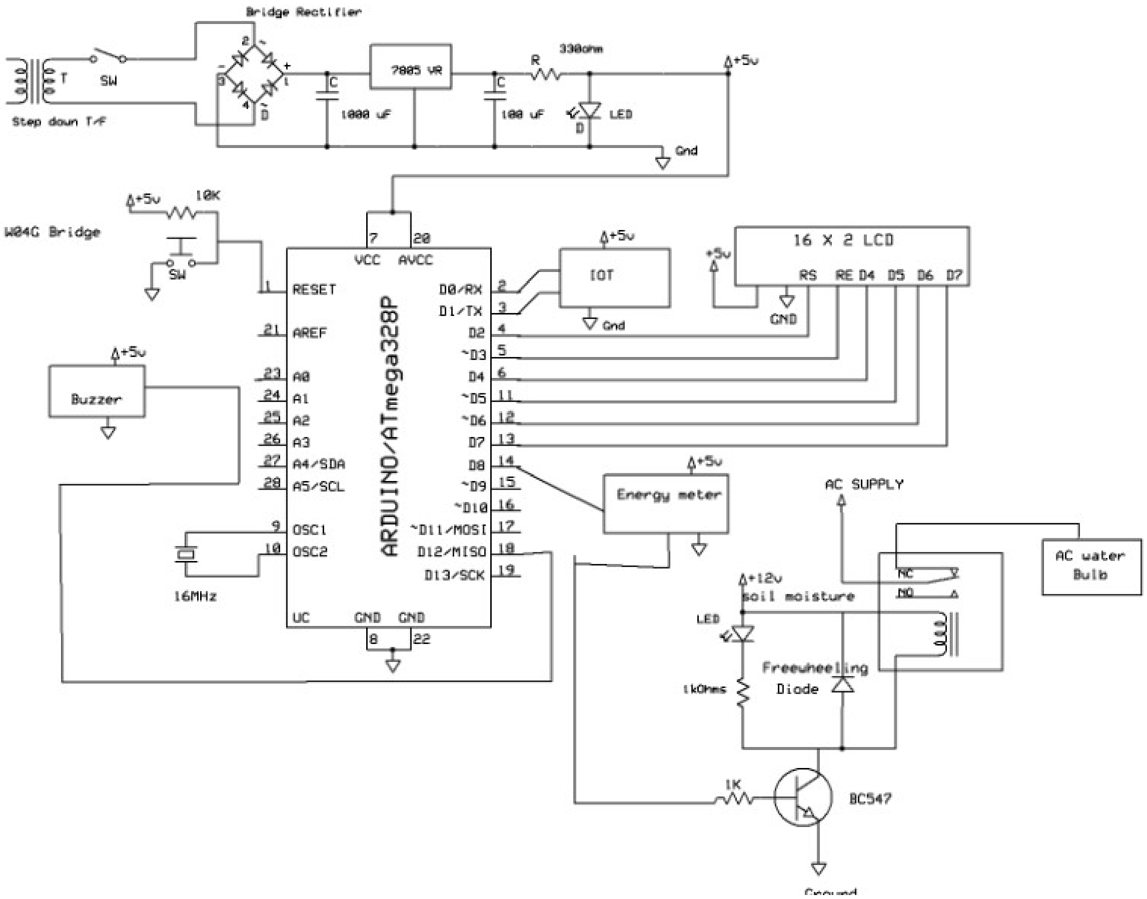
In 2010, using Multi-appliance power disaggregation technology implementers implemented the linear detection algorithm to determine which appliances are active in their power contributions. Problems are robust to errors in this database. [6] In 2011, using cloud computing technology found the solution for efficiency calculation of individual equipment. [10] In 2012, using three feedback system, monitored the energy in residential Real-Time. It is critical to the continuing engagement and use of the device to save energy. Residences to determine the feedback provided by real-time energy monitors results in lower residential consumption rates during the 30 days after installation. [8] In 2013, using GREEN technology is the smallest Zigbee-compatible node in existence. This technology will possible in every place sensing of a different data types, from energy metering to environmental monitoring. [9] 1n 2014, GSM technology implemented automatic power will be reading. [11] In 2016. Using wifi technology application can develop for Apple and BlackBerry 10 OS, thus providing multiple platform users support [2] In 2017, using IOT technology An IoT device was created for measuring the voltage, current, power and energy of a three-phase four-line power line in a laboratory building [12] Through a brief review of the published literature and previously done work, we can say that the researches have done a severe work on the plc power line communication and Internet of Things (IoT). It is concluded from the ken study of their work that in today`s world PLC & IoT based meter could improve the overall efficiency of the existing or present system and could help in examining the unnecessary losses of power in different areas.

CHAPTER 2: PROPOSED SYSTEM



In the proposed method, the consumer can manage their energy consumption by knowing their energy usage time to time. This method not only provides two way communications between utility and consumer but also provides other functions that are if the consumer fails to pay the electricity bill the energy supply would be cut down from the utility side and once the bill is paid the energy supply is reconnected. Another huge advantage of this system is that it notifies the consumer & utility at the event of the meter tampering. By this information the consumer & utility can control the tampering are reduce energy crises

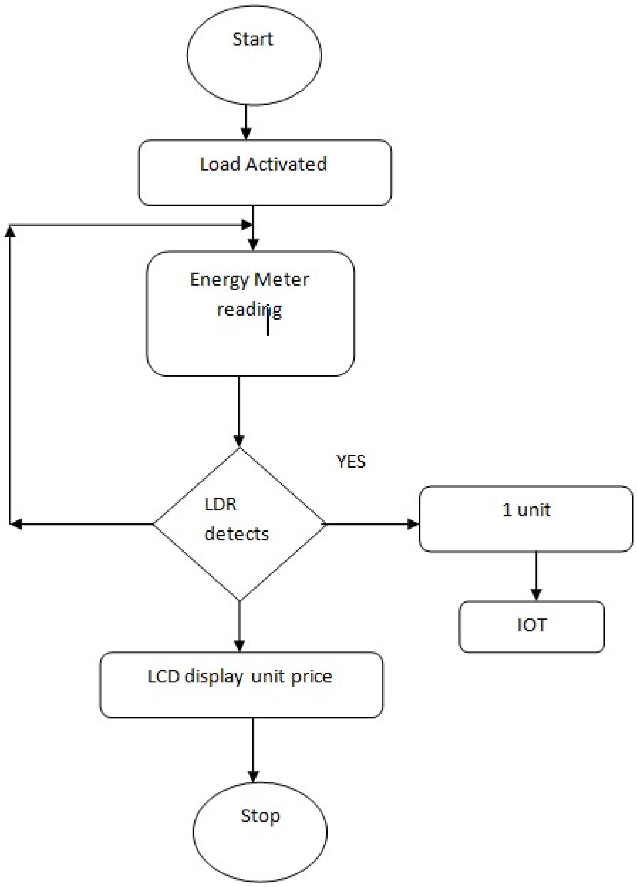
When the various appliances of the household consume energy the energy meter reads the reading continuously and this consumed load can be seen on meter. We can see that the LED on meter continuously• blinks which counts the meter reading. Based on The blinking, the units are counted. Normally,• 3200 blinks is one unit. In our project we are trying to develop, a system• in which Arduino Uno act as main controller, which continuously monitor energy meter. As per the blinking of LED on energy meter the• Arduino will measure the unit consumption. The measured reading with the calculation of• the cost will be continuously displayed on web page that we have designed. Threshold value can be set on webpage with the• help of Wi-Fi, as per the consumer’s requirement. When the consumers reading will be near about to the set threshold value it will send a notification value to the consumer. This threshold value notification will increase• the awareness amongst the consumer about the energy. When the consumer gets the notification he can• visit the webpage and change the threshold value. If the consumer is not aware with the threshold• notification, then the meter will automatically get off. Then the consumer has to visit the webpage again and increment the threshold value. By the incrementation, the meter will automatically get ON. Finally the overall monthly bill with cost will be• sent to customer as well as service provider in the form of text at first day of every month.



SCHEMATIC

WORKING

The smart meter will monitor by using Arduino nano microcontroller that is ATMEGA328. It maintains 8bit data size, operating range will be 3.3v to 5v. Wi-Fi module (ESP8266) works under six AT commands. Interfacing the Wi-Fi module, liquid crystal display, buzzer, and meter pulse by using C language on Arduino ID1.6.9. LCD is 2line 16 characters, here providing 5v to activate and then it displays the IP address which needs to connect the Wi-Fi module to send the data to processor. The crystal oscillator is used to convert the digital current signals to alternate current signal which requires maintaining the entire module of energy monitoring system. Load takes 5v power from the power transformer. Energy meter will read the pulse to calculate the amount of consumed power. Here meter pulse will be counted for calculating how much power is consumed by the consumer. One example to calculate the amount for consumed power.



CHAPTER 5
HARDWARE DESCRIPTION

2.1 Arduino:

Figure 2.1 ATMEGA 328 Microcontrollers

2.2 Pin Diagram:

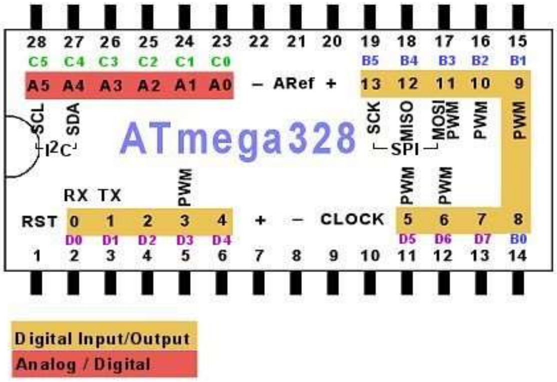


Figure 2.2 ATMEGA 328 PIN diagram

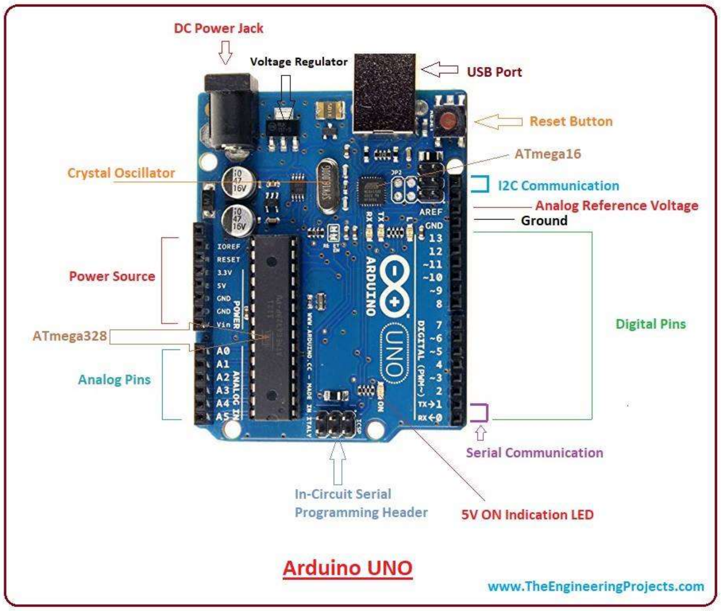


Figure 2.3 ARDUINO Development Board

VCC:

Digital supply voltage magnitude of the voltage range between 4.5 to 5.5 V for the ATmega8 and 2.7 to 5.5 V for ATmega8L

GND:

Ground Zero reference digital voltage supply.

PORTB (PB7.. PB0) :

PORTB is a port I / O two-way (bidirectional) 8-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTB pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

PORTC (PC5.. PC0) :

PORTC is a port I / O two-way (bidirectional) 7-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTC pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

PC6/RESET:

If RSTDISBL Fuse programmed, PC6 then serves as a pin I / O but with different characteristics. PC0 to PC5 If Fuse RSTDISBL not programmed, then serves as input Reset PC6. LOW signal on this pin with a minimum width of 1.5 microseconds will bring the microcontroller into reset condition, although the clock is not running.

PORTD (PD7.. PD0) :

PORTD is a port I / O two-way (bidirectional) 8-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTD pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

2.3 Architecture:

Memory: It has 8 Kb of Flash program memory (10,000 Write/Erase cycles durability), 512 Bytes of EEPROM (100,000 Write/Erase Cycles). 1Kbyte Internal SRAM

I/O Ports: 23 I/ line can be obtained from three ports; namely Port B, Port C and Port D.

Interrupts: Two External Interrupt source, located at port D. 19 different interrupt vectors supporting 19 events generated by internal peripherals.

Timer/Counter: Three Internal Timers are available, two 8 bit, one 16 bit, offering various operating modes and supporting internal or external clocking.

SPI (Serial Peripheral interface): ATmega8 holds three communication devices integrated. One of them is Serial Peripheral Interface. Four pins are assigned to Atmega8 to implement this scheme of communication.

USART: One of the most powerful communication solutions is USART and ATmega8 supports both synchronous and asynchronous data transfer schemes. It has three pins assigned for that. In many projects, this module is extensively used for PC-Micro controller communication.

TWI (Two Wire Interface): Another communication device that is present in ATmega8 is Two Wire Interface. It allows designers to set up a commutation between two devices using just two wires along with a common ground connection, As the TWI output is made by means of open collector outputs, thus external pull up resistors are required to make the circuit.

Analog Comparator: A comparator module is integrated in the IC that provides comparison facility between two voltages connected to the two inputs of the Analog comparator via External pins attached to the micro controller.

Analog to Digital Converter: Inbuilt analog to digital converter can convert an analog input signal into digital data of 10bit resolution. For most of the low end application, this much resolution is enough.

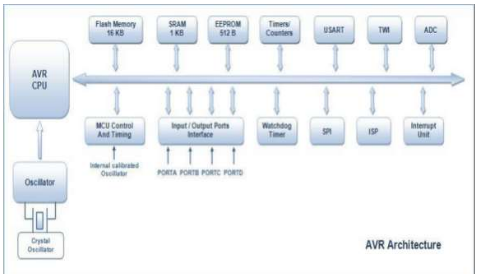


Figure 2.4 AVR Architecture

Features:

• High-performance, Low-power Atmel®AVR® 8-bit Microcontroller

• Advanced RISC Architecture

– 130 Powerful Instructions – Most Single-clock Cycle Execution

– 32 × 8 General Purpose Working Registers

– Fully Static Operation

– Up to 16MIPS Throughput at 16MHz

– On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory segments

– 8Kbytes of In-System Self-programmable Flash program memory

– 512Bytes EEPROM

– 1Kbyte Internal SRAM

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

– Data retention: 20 years at 85°C/100 years at 25°C(1)

– Optional Boot Code Section with Independent Lock Bits

In-System Programming by On-chip Boot Program

True Read-While-Write Operation

– Programming Lock for Software Security

• Peripheral Features

– Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode

– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

– Real Time Counter with Separate Oscillator

– Three PWM Channels

– 8-channel ADC in TQFP and QFN/MLF package

Eight Channels 10-bit Accuracy

– 6-channel ADC in PDIP package

Six Channels 10-bit Accuracy

– Byte-oriented Two-wire Serial Interface

– Programmable Serial USART

– Master/Slave SPI Serial Interface

– Programmable Watchdog Timer with Separate On-chip Oscillator

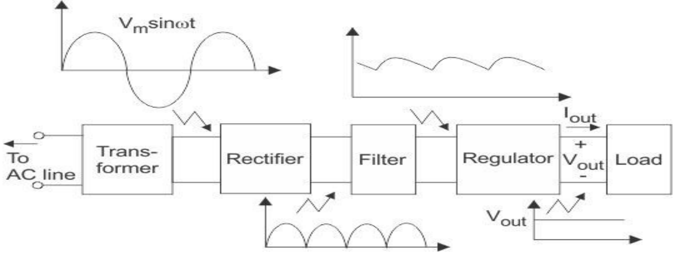
– On-chip Analog Comparator

• Special Microcontroller Features
– Power-on Reset and Programmable Brown-out Detection
– Internal Calibrated RC Oscillator
– External and Internal Interrupt Sources
– Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby

• I/O and Packages
– 23 Programmable I/O Lines
– 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
• Operating Voltages
– 2.7V - 5.5V (ATmega8L)
– 4.5V - 5.5V (ATmega8)
• Speed Grades
– 0 - 8MHz (ATmega8L)
– 0 - 16MHz (ATmega8)
• Power Consumption at 4Mhz, 3V, 25oC
– Active: 3.6mA
– Idle Mode: 1.0mA
– Power-down Mode: 0.5µA

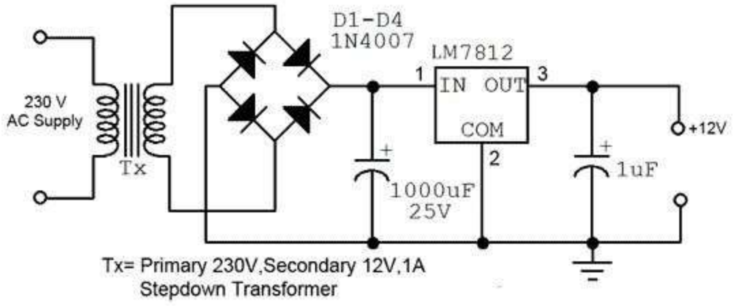
Power Supply:

A regulated power supply converts unregulated AC (Alternating Current) to a constant DC (Direct Current). A regulated power supply is used to ensure that the output remains constant even if the input changes. A regulated DC power supply is also known as a linear power supply; it is an embedded circuit and consists of various blocks. The regulated power supply will accept an AC input and give a constant DC output. The figure below shows the block diagram of a typical regulated DC power supply.



Components of typical linear power supply

Figure 2.8 Block diagram of power supply



Tx= Primary 230V,Secondary 12V,1A
Stepdown Transformer

Figure 2.9 Power supply circuit diagram

3.5 ESP8266 MODULE

1 Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) of high performance wireless SOCs, for mobile platform designers, provides unsurpassed ability to embed Wi-Fi capabilities within other systems, at the lowest cost with the greatest functionality

2 Technology Overview

ESP8266 offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

When ESP8266 hosts the application, and when it is the only application processor in the device, it is able to boot up directly from an external flash. It has integrated cache to improve the performance of the system in such applications, and to minimize the memory requirements.

Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any microcontroller-based design with simple connectivity through UART interface or the CPU AHB bridge interface.

ESP8266 on-board processing and storage capabilities allow it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. With its high degree of on-chip integration, which includes the antenna switch balun, power management converters, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

Sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

3 Features

- 802.11 b/g/n protocol
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- +19.5dBm output power in 802.11b mode
- Integrated temperature sensor

- Supports antenna diversity
- Power down leakage current of < 10uA
- Integrated low power 32-bit CPU could be used as application processor

ESP8266 Applications

- Smart power plugs
- Home automation
- Mesh network
- Industrial wireless control

Specifications

7.1 Current Consumption

The following current consumption is based on 3.3V supply, and 25□C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitter's measurements are based on 90% duty cycle, continuous transmit mode.

Mode		Typ	Unit
Transmit 802.11b, CCK 1Mbps, POUT=+19.5dBm		215	mA
Transmit 802.11b, CCK 11Mbps, POUT=+18.5dBm		197	mA
Transmit 802.11g, OFDM 54Mbps, POUT =+16dBm		145	mA
Transmit 802.11n, MCS7, POUT=+14dBm		135	mA
Receive 802.11b, packet length=1024 byte, -80dBm		60	mA
Receive 802.11g, packet length=1024 byte, -70dBm		60	mA
Receive 802.11n, packet length=1024 byte, -65dBm		62	mA
Standby		0.9	mA
Deep sleep		10	uA
Power save mode DTIM 1		1.2	mA
Power save mode DTIM 3		0.86	mA
Total shutdown		0.5	uA

CPU, Memory and Interfaces

CPU

This chip embeds an ultra low power Micro 32-bit CPU, with 16-bit thumb mode. This CPU can be interfaced using:

- code RAM/ROM interface (iBus) that goes to the memory controller, that can also be used to access external flash memory,
- data RAM interface (dBus), that also goes to the memory controller
- AHB interface, for register access, and
- JTAG interface for debugging

Memory Controller

The memory controller contains ROM, and SRAM. It is accessed by the CPU using the iBus, dBus and AHB interface. Any of these interfaces can request access to the ROM or RAM modules, and the memory controller arbiters serve these 3 interfaces on a first-come-first-serve basis.

AHB and AHB Blocks

The AHB blocks performs the function of an arbiter, controls the AHB interfaces from the MAC, SDIO (host) and CPU. Depending on the address, the AHB data requests can go into one of the two slaves:

Data requests to the memory controller are usually high speed requests, and requests to the APB block are usually register access.

The APB block acts as a decoder. It is meant only for access to programmable registers within ESP8266's main blocks. Depending on the address, the APB request can go to the radio, SI/SPI, SDIO (host), GPIO, UART, real-time clock (RTC), MAC or digital baseband.

Interfaces

The ESP8266 contains several analog and digital interfaces described in the following sections.

Master SI / SPI Control (Optional)

The master serial interface (SI) can operate in two, three or four-wire bus configurations to control the EEPROM or other I2C/SPI devices. Multiple I2C devices with different device addresses are supported by sharing the 2-wire bus.

Multiple SPI devices are supported by sharing the clock and data signals, using separate software controlled GPIO pins as chip selects.

The SPI can be used for controlling external devices such as serial flash memories, audio CODECs, or other slave devices. It is set up as a standard master SPI device with 3 different enable pins:

- SPI_EN0,
- SPI_EN1,
- SPI_EN2.

Both SPI master and SPI slave are supported with the latter being used as a host interface.

SPI_EN0 is used as an enable signal to an external serial flash memory for downloading patch code and/or MIB-data to the baseband in an embedded application. In a host based application, patch code and MIB-data can alternatively be downloaded via the host interface. This pin is active low and should be left open if not used.

General Purpose IO

There are up to 16 GPIO pins. They can be assigned to various functions by the firmware. Each GPIO can be configured with internal pull-up/down, input available for sampling by a software register, input triggering an edge or level CPU interrupt, input triggering a level wakeup interrupt, open-drain or push-pull output driver, or output source from a software register, or a sigma-delta PWM DAC.

These pins are multiplexed with other functions such as host interface, UART, SI, Bluetooth coexistence, etc.

Digital IO Pads

The digital IO pads are bidirectional, non-inverting and tri-state. It includes input and an output buffer with tristate control inputs. Besides this, for low power operations, the IO can also be set to hold. For instance, when we power down the chip, all output enable signals can be set to hold low. Optional hold functionality can be built into the IO if requested. When the IO is not driven by the internal or external circuitry, the hold functionality can be used to hold the state to the last used state.

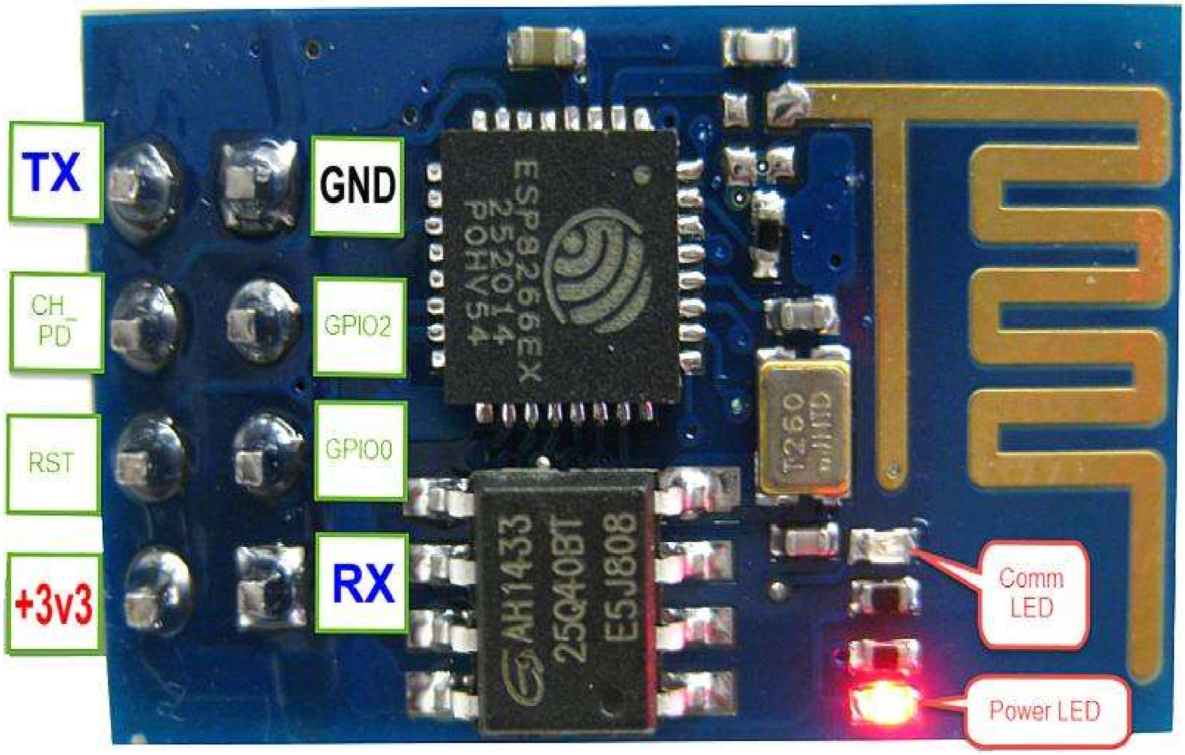
The hold functionality introduces some positive feedback into the pad. Hence, the external driver that drives the pad must be stronger than the positive feedback. The required drive strength is however small – in the range of 5uA.

Parameter	**Symbol**	**Min**	**Max**	**Unit**
Input low voltage	**VIL**	**-0.3**	**0.25□VIO**	**V**
Input high voltage	**VIH**	**0.75□VIO**	**3.6**	**V**
Input leakage current		**IIL**	**50**	**nA**
Output low voltage		**VOL**	**0.1□VIO**	**V**
Output high voltage		**VOH**	**0.8□VIO**	**V**
Input pin capacitance		**Cpad**	**2**	**pF**
VDDIO	**VIO**	**1.7**	**3.6**	**V**
Maximum drive capability		**IMAX**	**12**	**mA**
Temperature	**Tamb**	**-20**	**100**	□C

Applications

- **Wi-Fi Smart Hardware Converted from UART Serial Ports**
- **Sensor**
- **Smart Light**
- **Smart Plug**

Wi-Fi (Short for Wireless Fidelity) is a wireless technology that uses radio frequency to transmit data through the air. Wi-Fi has initial speeds of 1mbps to 2mbps. Wi-Fi transmits data in the frequency band of 2.4 GHz. It implements the concept of frequency division multiplexing technology. Range of Wi-Fi technology is 40-300 feet.



2.2 Wi-fi module

ESP8266 offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

When ESP8266 hosts the application, and when it is the only application processor in the device, it is able to boot up directly from an external flash. It has integrated cache to improve the performance of the system in such applications, and to minimize the memory requirements.

Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any microcontroller-based design with simple connectivity through UART interface or the CPU AHB bridge interface.

ESP8266 on-board processing and storage capabilities allow it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. With its high degree of on-chip integration, which includes the antenna switch balun, power management converters, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

Sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

2.2.1FEATURES

- 802.11 b/g/n protocol
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- +19.5dBm output power in 802.11b mode
- Integrated temperature sensor
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

2.2.2 ESP8266 APPLICATIONS

- Smart power plugs
- Home automation
- Mesh network
- Industrial wireless control
- Baby monitors
- IP Cameras
- Sensor networks

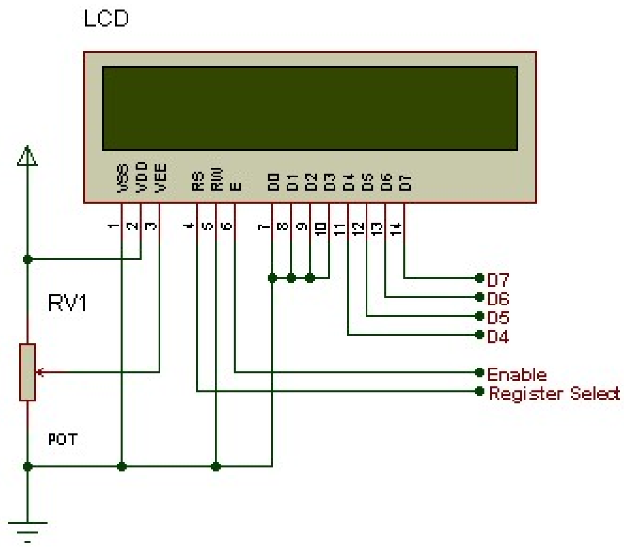
3.7 LCD DISPLAY

LCD Background:

One of the most common devices attached to a micro controller is an LCD display. Some of the most common LCD’s connected to the many microcontrollers are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Basic 16x 2 Characters LCD

Figure 1: LCD **Pin diagram**



Pin description:

Pin No.	Name	Description
Pin no. 1	**VSS**	Power supply (GND)
Pin no. 2	**VCC**	Power supply (+5V)
Pin no. 3	**VEE**	Contrast adjust
Pin no. 4	**RS**	0 = Instruction input 1 = Data input
Pin no. 5	**R/W**	0 = Write to LCD module 1 = Read from LCD module
Pin no. 6	**EN**	Enable signal
Pin no. 7	**D0**	Data bus line 0 (LSB)
Pin no. 8	**D1**	Data bus line 1
Pin no. 9	**D2**	Data bus line 2
Pin no. 10	**D3**	Data bus line 3
Pin no. 11	**D4**	Data bus line 4

Pin no. 12	**D5**	Data bus line 5
Pin no. 13	**D6**	Data bus line 6
Pin no. 14	**D7**	Data bus line 7 (MSB)

Table 1: Character LCD **pins with Microcontroller**

The LCD requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

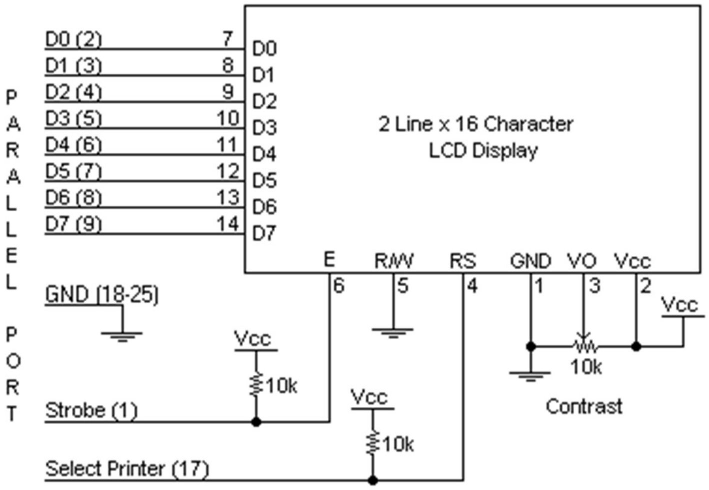
The **EN** line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring **EN** high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The **RS** line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen we would set RS high.

The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Schematic:



Circuit Description:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. We can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if we have trouble with the circuit working properly.

SETB RW

Handling the EN control line:

As we mentioned above, the EN line is used to tell the LCD that we are ready for it to execute an instruction that we've prepared on the data bus and on the other control lines. Note that the EN line must be raised/ lowered before/after each instruction sent to the LCD regardless of whether that instruction is read or write text or instruction. In short, we must always manipulate EN when communicating with the LCD. EN is the LCD's way of knowing that we are talking to it. If we don't raise/lower EN, the LCD doesn't know we're talking to it on the other lines.

Thus, before we interact in any way with the LCD we will always bring the **EN** line low with the following instruction:

Checking the busy status of the LCD:

As previously mentioned, it takes a certain amount of time for each instruction to be executed by the LCD. The delay varies depending on the frequency of the crystal attached to the oscillator input of the LCD as well as the instruction which is being executed.

While it is possible to write code that waits for a specific amount of time to allow the LCD to execute instructions, this method of "waiting" is not very flexible. If the crystal frequency is changed, the software will need to be modified. A more robust method of programming is to use the "Get LCD Status" command to determine whether the LCD is still busy executing the last instruction received.

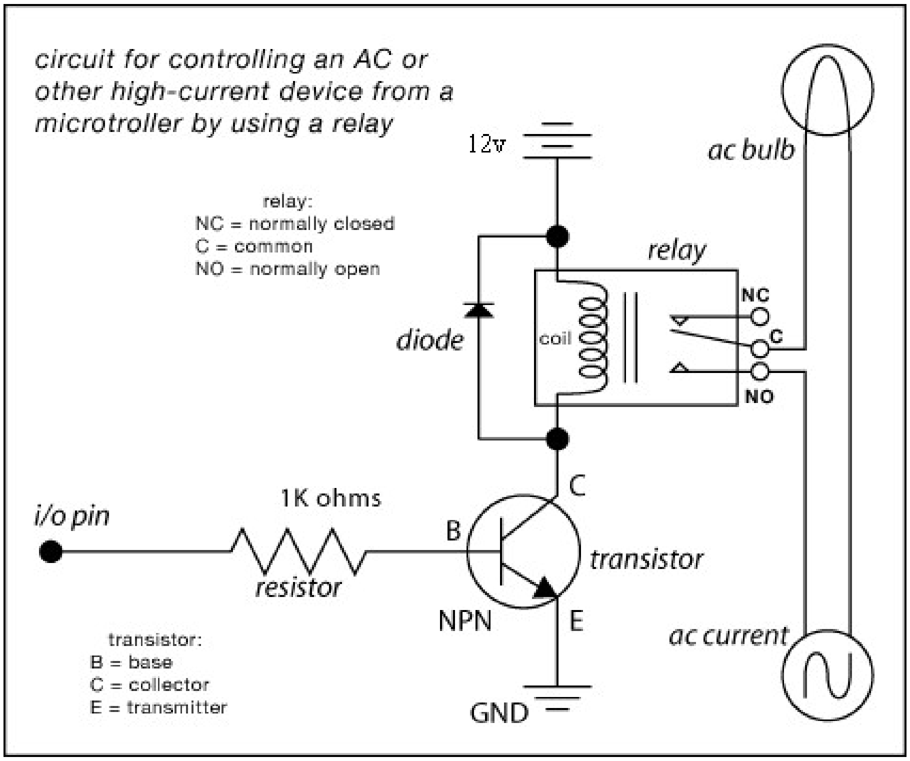
The "Get LCD Status" command will return to us two tidbits of information; the information that is useful to us right now is found in DB7. In summary, when we issue the "Get LCD Status" command the LCD will immediately raise DB7 if it's still busy executing a command or lower DB7 to indicate that the LCD is no longer occupied. Thus our program can query the LCD until DB7 goes low, indicating the LCD is no longer busy. At that point we are free to continue and send the next command.

Applications:

- Medical equipment
- Electronic test equipment
- Industrial machinery Interface
- Serial terminal
- Advertising system
- EPOS
- Restaurant ordering systems
- Gaming box
- Security systems
- R&D Test units
- Climatizing units
- PLC Interface
- Simulators
- Environmental monitoring
- Lab development
- Student projects
- Home automation
- PC external display
- HMI operator interface.

3.8 Relay:

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism, but other operating principles are also used. Relays find applications where it is necessary to control a circuit by a low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays found extensive use in telephone exchanges and early computers to perform logical operations. A type of relay that can handle the high power required to directly drive an electric motor is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device triggered by light to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protection relays".



3.5 ENERGY METER:

An **electric meter** or **energy meter** is a device that measures the amount of electrical energy consumed by a residence, business, or an electrically-powered device.

Electric meters are typically calibrated in billing units, the most common one being the kilowatt hour. Periodic readings of electric meters establish billing cycles and energy used during a cycle.

In settings when energy savings during certain periods are desired, meters may measure demand, the maximum use of power in some interval. In some areas, the electric rates are higher during certain times of day, to encourage reduction in use. Also, in some areas meters have relays to turn off nonessential equipment

3.5.1 Theory:

The first accurate, recording electricity consumption meter was a DC meter by Dr Hermann Aron, who patented it in 1883. Hugo Hirst of the General Electric Company introduced it commercially into Great Britain from 1888.[2] Meters had been used prior to this, but they measured the rate of power consumption at that particular moment. Aron's meter recorded the total energy used over time, and showed it on a series of clock dials.

The prototype of the world's first kilowatt-hour meter, inventor: Ottó Bláthy. Made in Ganz Company, Hungary in 1889

The first specimen of the AC kilowatt-hour meter produced on the basis of Hungarian Ottó Bláthy's patent and named after him was presented by the Ganz Works at the Frankfurt Fair in the autumn of 1889, and the first induction kilowatt-hour meter was already marketed by the factory at the end of the same year. These were the first alternating-current wattmeter's, known by the name of Bláthy-meters

Unit of measurement

Panel-mounted solid state electricity meter, connected to a 2 MVA electricity substation. Remote current and voltage sensors can be read and programmed remotely by modem and locally by infra-red. The circle with two dots is the infra-red port. Tamper-evident seals can be seen.

The most common unit of measurement on the electricity meter is the kilowatt hour, which is equal to the amount of energy used by a load of one kilowatt over a period of one hour, or 3,600,000 joules. Some electricity companies use the SI mega joule instead.

Demand is normally measured in watts, but averaged over a period, most often a quarter or half hour.

Reactive power is measured in "Volt-amperes reactive", (varh) in kilovar-hours. By convention, a "lagging" or inductive load, such as a motor, will have positive reactive power. A "leading", or capacitive load, will have negative reactive power.[4]

Volt-amperes measures all power passed through a distribution network, including reactive and actual. This is equal to the product of root-mean-square volts and amperes.

Distortion of the electric current by loads is measured in several ways. Power factor is the ratio of resistive (or real power) to volt-amperes. A capacitive load has a leading power factor, and an inductive load has a lagging power factor. A purely resistive load (such as a filament lamp, heater or kettle) exhibits a power factor of 1. Current harmonics are a measure of distortion of the wave form. For example, electronic loads such as computer power supplies draw their current at the voltage peak to fill their internal storage elements. This can lead to a significant voltage drop near the supply voltage peak which shows as a flattening of the voltage waveform. This flattening causes odd harmonics which are not permissible if they exceed specific limits, as they are not only wasteful, but may interfere with the operation of other equipment. Harmonic emissions are mandated by law in EU and other countries to fall within specified limits.

CHAPTER 4: SOFTWARE DESCRIPTION

This project is implemented using following software's:

- Express PCB – for designing circuit
- ARDUINO

4.1 Express PCB:

Breadboards are great for prototyping equipment as it allows great flexibility to modify a design when needed; however the final product of a project, ideally should have a neat PCB, few cables, and survive a shake test. Not only is a proper PCB neater but it is also more durable as there are no cables which can yank loose.

Express PCB is a software tool to design PCBs specifically for manufacture by the company Express PCB (no other PCB maker accepts Express PCB files). It is very easy to use, but it does have several limitations.

It can be likened to more of a toy then a professional CAD program.

It has a poor part library (which we can work around)

It cannot import or export files in different formats

It cannot be used to make prepare boards for DIY production

Express PCB has been used to design many PCBs (some layered and with surface-mount parts. Print out PCB patterns and use the toner transfer method with an Etch Resistant Pen to make boards. However, Express PCB does not have a nice print layout. Here is the procedure to design in Express PCB and clean up the patterns so they print nicely.

4.1.1 Preparing Express PCB for First Use:

Express PCB comes with a less then exciting list of parts. So before any project is started head over to Audio logical and grab the additional parts by morsel, ppl, and tangent, and extract them into your Express PCB directory. At this point start the program and get ready to setup the workspace to suit your style.

Click View -> Options. In this menu, setup the units for “mm” or “in” depending on how you think, and click “see through the top copper layer” at the bottom. The standard color scheme of red and green is generally used but it is not as pleasing as red and blue.

4.1.2 The Interface:

When a project is first started you will be greeted with a yellow outline. This yellow outline is the dimension of the PCB. Typically after positioning of parts and traces, move them to their final position and then crop the PCB to the correct size. However, in designing a board with a certain size constraint, crop the PCB to the correct size before starting.

Fig: 4.1 show the toolbar in which the each button has the following functions:

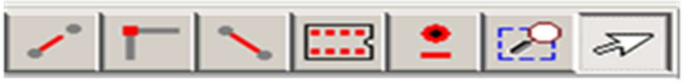


Fig 4.1: Tool bar necessary for the interface

- The select tool: It is fairly obvious what this does. It allows you to move and manipulate parts. When this tool is selected the top toolbar will show buttons to move traces to the top / bottom copper layer, and rotate buttons.
- The zoom to selection tool: does just that.
- The place pad: button allows you to place small soldier pads which are useful for board connections or if a part is not in the part library but the part dimensions are available. When this

tool is selected the top toolbar will give you a large selection of round holes, square holes and surface mount pads.

.

4.1.3 Design Considerations:

Before starting a project there are several ways to design a PCB and one must be chosen to suit the project’s needs.

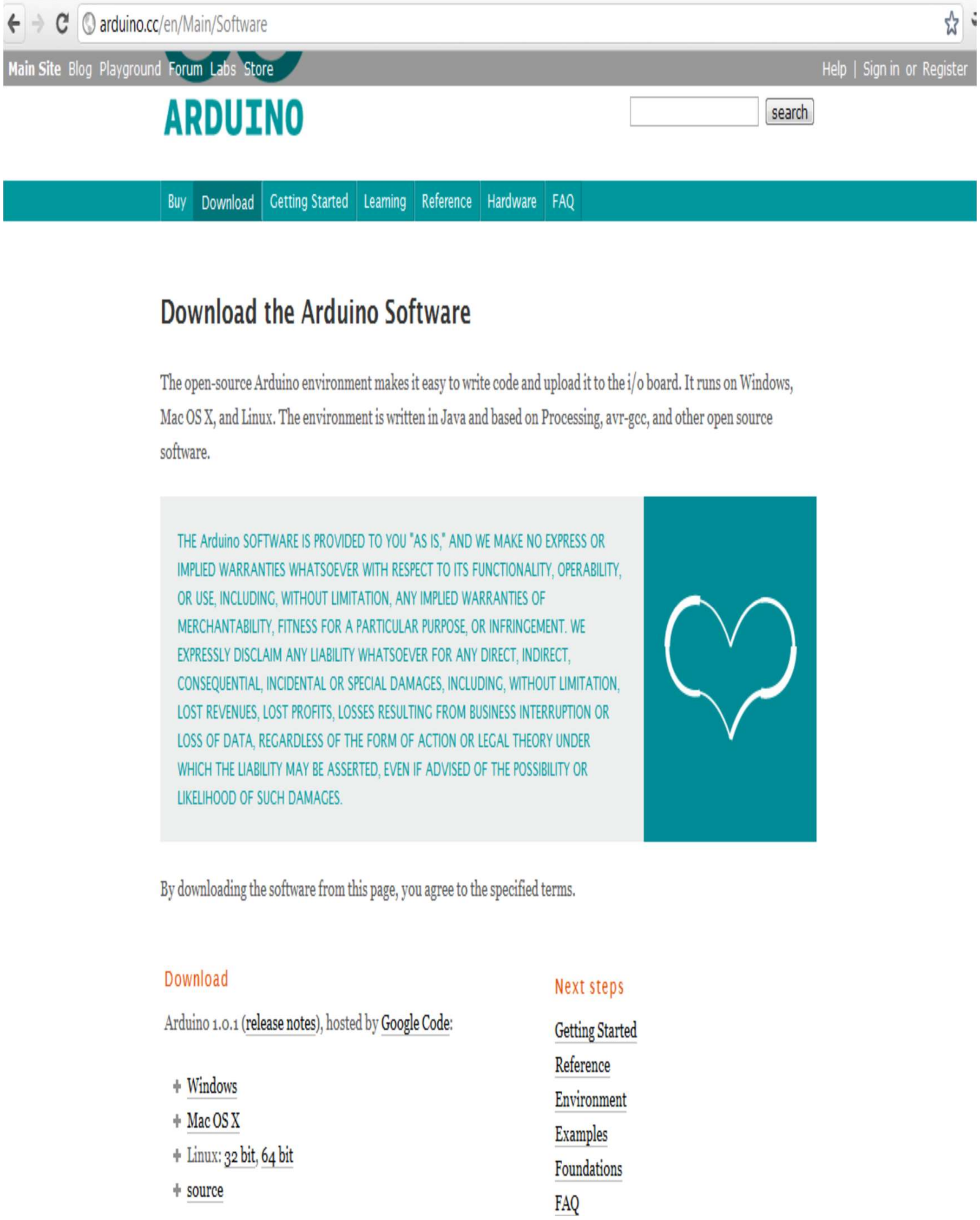
Single sided, or double sided?

When making a PCB you have the option of making a single sided board, or a double sided board. Single sided boards are cheaper to produce and easier to etch, but much harder to design for large projects. If a lot of parts are being used in a small space it may be difficult to make a single sided board without jumpering over traces with a cable. While there’s technically nothing wrong with this, it should be avoided if the signal travelling over the traces is sensitive (e.g. audio signals).

A double sided board is more expensive to produce professionally, more difficult to etch on a DIY board, but makes the layout of components a lot smaller and easier. It should be noted that if a trace is running on the top layer, check with the components to make sure you can get to its pins with a soldering iron. Large capacitors, relays, and similar parts which don’t have axial leads can NOT have traces on top unless boards are plated professionally.

Ground-plane or other special purposes for one side

Some projects like power supplies or amps can benefit from having a solid plane to use for ground. In power supplies this can reduce noise, and in amps it minimizes the distance between parts and their ground connections, and keeps the ground signal as simple as possible. However, care must be taken with stubborn chips such as the TPA6120 amplifier from TI. The TPA6120 datasheet specifies not to run a ground plane under the pins or signal traces of this chip as the capacitance generated could effect performance negatively.


arduino.cc/en/Main/Software
Main Site Blog Playground Forum Labs Store
Help | Sign in or Register
ARDUINO
search
Buy Download Getting Started Learning Reference Hardware FAQ

Download the Arduino Software

The open-source Arduino environment makes it easy to write code and upload it to the i/o board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing, avr-gcc, and other open source software.

THE Arduino SOFTWARE IS PROVIDED TO YOU "AS IS," AND WE MAKE NO EXPRESS OR IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCTIONALITY, OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR INFRINGEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEVER FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, LOSSES RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, REGARDLESS OF THE FORM OF ACTION OR LEGAL THEORY UNDER WHICH THE LIABILITY MAY BE ASSERTED, EVEN IF ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

By downloading the software from this page, you agree to the specified terms.

Download

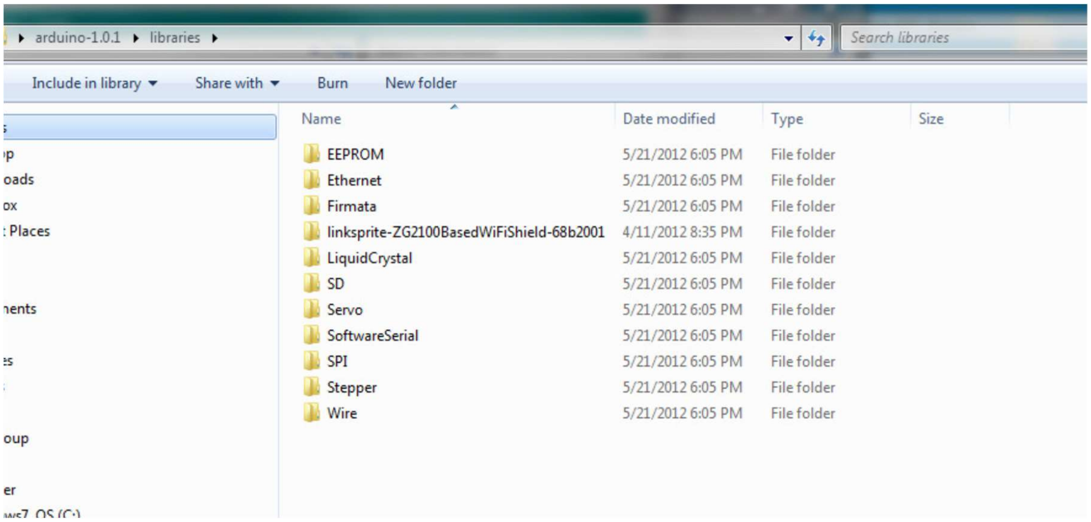
Arduino 1.0.1 (release notes), hosted by Google Code:

- Windows
- Mac OS X
- Linux: 32 bit, 64 bit
- source

Next steps

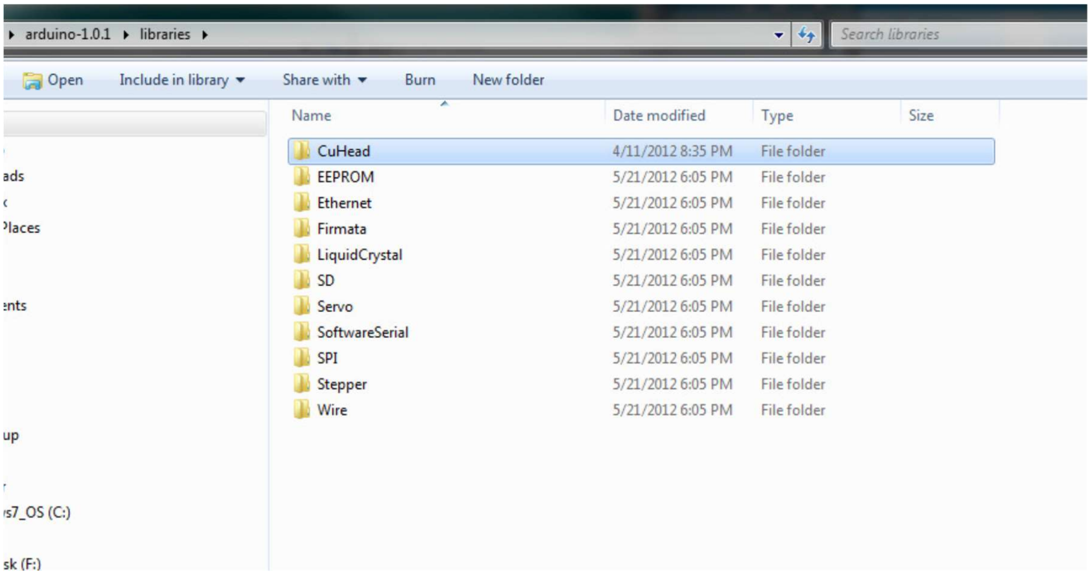
Getting Started
Reference
Environment
Examples
Foundations
FAQ

In next step download library



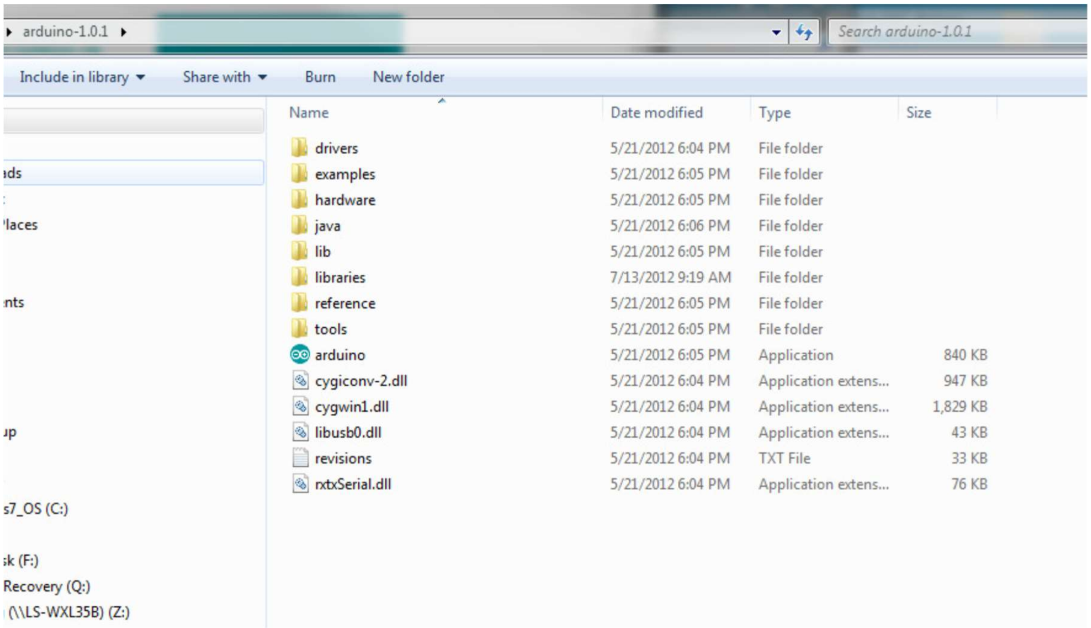
Name	Date modified	Type	Size
EEPROM	5/21/2012 6:05 PM	File folder	
Ethernet	5/21/2012 6:05 PM	File folder	
Firmata	5/21/2012 6:05 PM	File folder	
linksprite-ZG2100BasedWiFiShield-68b2001	4/11/2012 8:35 PM	File folder	
LiquidCrystal	5/21/2012 6:05 PM	File folder	
SD	5/21/2012 6:05 PM	File folder	
Servo	5/21/2012 6:05 PM	File folder	
SoftwareSerial	5/21/2012 6:05 PM	File folder	
SPI	5/21/2012 6:05 PM	File folder	
Stepper	5/21/2012 6:05 PM	File folder	
Wire	5/21/2012 6:05 PM	File folder	

As Arduino doesn’t recognize the directory name, please rename it



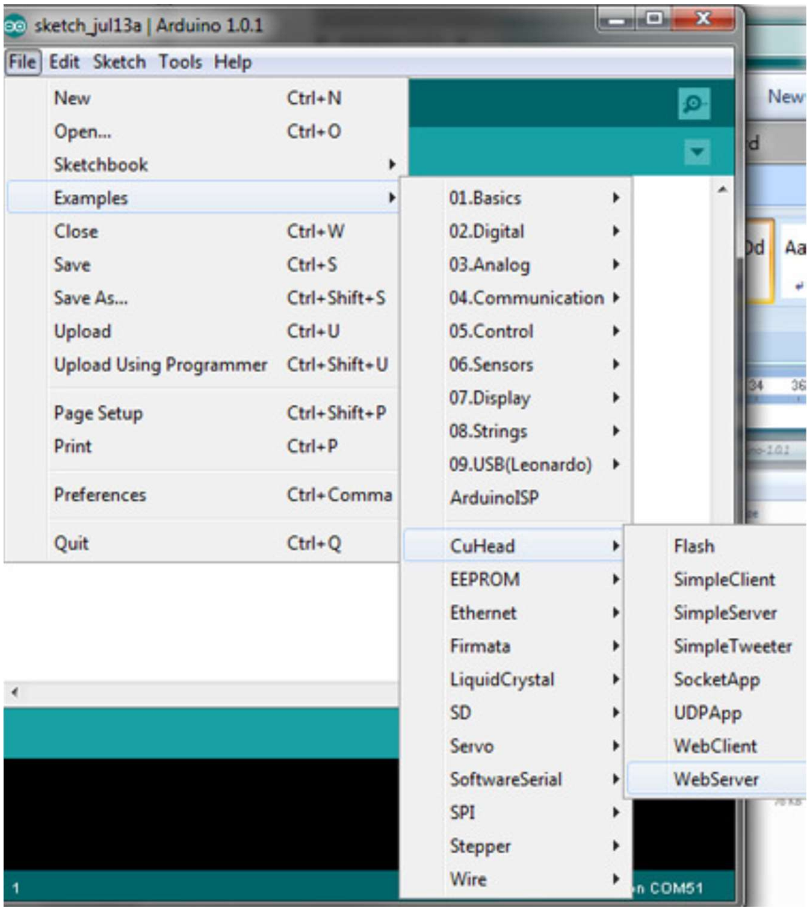
Name	Date modified	Type	Size
CuHead	4/11/2012 8:35 PM	File folder	
EEPROM	5/21/2012 6:05 PM	File folder	
Ethernet	5/21/2012 6:05 PM	File folder	
Firmata	5/21/2012 6:05 PM	File folder	
LiquidCrystal	5/21/2012 6:05 PM	File folder	
SD	5/21/2012 6:05 PM	File folder	
Servo	5/21/2012 6:05 PM	File folder	
SoftwareSerial	5/21/2012 6:05 PM	File folder	
SPI	5/21/2012 6:05 PM	File folder	
Stepper	5/21/2012 6:05 PM	File folder	
Wire	5/21/2012 6:05 PM	File folder	

Launch Arduino by double click “arduino” below

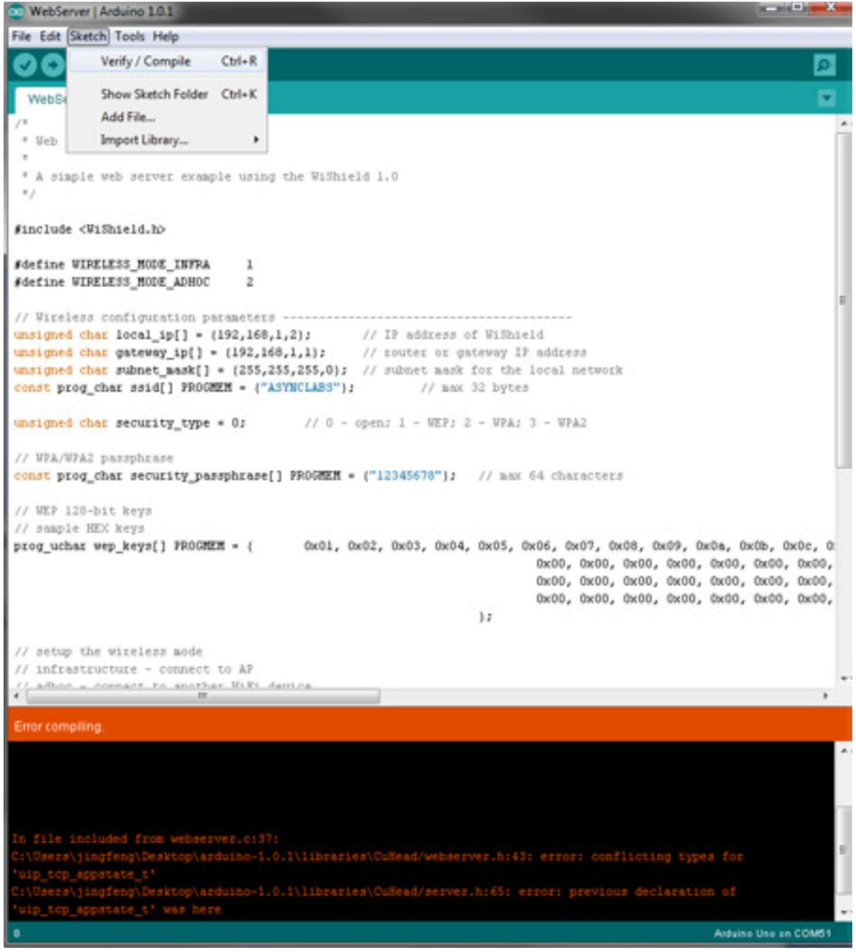


Name	Date modified	Type	Size
drivers	5/21/2012 6:04 PM	File folder	
examples	5/21/2012 6:05 PM	File folder	
hardware	5/21/2012 6:05 PM	File folder	
java	5/21/2012 6:06 PM	File folder	
lib	5/21/2012 6:05 PM	File folder	
libraries	7/13/2012 9:19 AM	File folder	
reference	5/21/2012 6:05 PM	File folder	
tools	5/21/2012 6:05 PM	File folder	
arduino	5/21/2012 6:05 PM	Application	840 KB
cygiconv-2.dll	5/21/2012 6:04 PM	Application extens...	947 KB
cygwin1.dll	5/21/2012 6:04 PM	Application extens...	1,829 KB
libusb0.dll	5/21/2012 6:04 PM	Application extens...	43 KB
revisions	5/21/2012 6:04 PM	TXT File	33 KB
rxtxSerial.dll	5/21/2012 6:04 PM	Application extens...	76 KB

One example



Select the target board as “Arduino Uno


WebServer | Arduino 1.0.1
File Edit Sketch Tools Help
Verify / Compile Ctrl+R
Show Sketch Folder Ctrl+K
Add File...
Import Library...
WebS

```
/*
 * Web
 *
 * A simple web server example using the WiShield 1.0
 */

#include <WiShield.h>

#define WIRELESS_MODE_INFRA     1
#define WIRELESS_MODE_ADHOC     2

// Wireless configuration parameters ----------------------------------------
unsigned char local_ip[] = {192,168,1,2};       // IP address of WiShield
unsigned char gateway_ip[] = {192,168,1,1};     // router or gateway IP address
unsigned char subnet_mask[] = {255,255,255,0};  // subnet mask for the local network
const prog_char ssid[] PROGMEM = {"ASYNCLABS"};         // max 32 bytes

unsigned char security_type = 0;        // 0 - open; 1 - WEP; 2 - WPA; 3 - WPA2

// WPA/WPA2 passphrase
const prog_char security_passphrase[] PROGMEM = {"12345678"};   // max 64 characters

// WEP 128-bit keys
// sample HEX keys
prog_uchar wep_keys[] PROGMEM = {       0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0
                                                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                                        };


// setup the wireless mode
// infrastructure - connect to AP
// adhoc - connect to another WiFi device
```

Error compiling.

```
In file included from webserver.c:37:
C:\Users\jingfeng\Desktop\arduino-1.0.1\libraries\CuHead/webserver.h:43: error: conflicting types for
'uip_tcp_appstate_t'
C:\Users\jingfeng\Desktop\arduino-1.0.1\libraries\CuHead/server.h:65: error: previous declaration of
'uip_tcp_appstate_t' was here
```

CHAPTER 6
CODE

```
#include <LiquidCrystal.h>
#include <stdio.h>

LiquidCrystal lcd(6, 7, 5, 4, 3, 2);

int mtr   = 9;
int relay = 8;

int buzzer = 13;
void beep()
{
  digitalWrite(buzzer, LOW);delay(1500);digitalWrite(buzzer, HIGH);
}
void okcheck()
{
  unsigned char rcr;
  do{
      rcr = Serial.read();
    }while(rcr != 'K');
}
void things_send()
{
unsigned char recr;
 Serial.write("AT+CIPMUX=1\r\n");delay(2000);
 Serial.write("AT+CIPSTART=4,\"TCP\",\"184.106.153.149\",80\r\n"); delay(4000);
//OK  LINKED
 Serial.write("AT+CIPSEND=4,78\r\n");  delay(3000);
 Serial.write("GET
https://api.thingspeak.com/update?api_key=0PHK2UK998YJ7DFU&");
}

void things_rcv()
{
unsigned char recr;
  Serial.write("AT+CIPSTART=4,\"TCP\",\"184.106.153.149\",80\r\n");  delay(4000);
  Serial.write("AT+CIPSEND=4,73\r\n");    delay(3000);
  Serial.write("GET https://api.thingspeak.com/channels/449069/fields/3.json?results=1");
}

void things_done()
{
 Serial.write("\r\n\r\n");   delay(4000);
}

int units=0;
int amount=0;
int perc=0;

void setup()
{
  Serial.begin(9600);//serialEvent();


  pinMode(buzzer, OUTPUT);  pinMode(relay, OUTPUT);

  digitalWrite(buzzer, HIGH);digitalWrite(relay, LOW);


  lcd.begin(16, 2);
  lcd.print("IOT Energymeter");
  delay(1500);
   wifiinit();
  lcd.clear();
    lcd.setCursor(0, 0);   //column,row
    lcd.print("U:");          //2,0
    lcd.setCursor(0, 1);   //column,row
    lcd.print("A:");          //2,1
 digitalWrite(relay, HIGH);delay(1000);
}
void loop()
{
  if(digitalRead(mtr) == LOW)
    {
      units++;
      amount = (units * 3);
perc = (units * 10);
lcd.setCursor(3,0);convertl(units);
```

```
   lcd.setCursor(11,0);convertl(perc);lcd.print("%");
  lcd.setCursor(3,1);convertl(amount);
   if(units >= 10)
     {
      digitalWrite(relay, LOW);
      beep();
     }

   lcd.setCursor(13,1);lcd.print("1   ");
     things_send();
     Serial.write("field3=");
     converts(units);
     things_done();
     lcd.setCursor(13,1);lcd.print("    ");

   for(cntl=0;cntl<35;cntl++)
      {lcd.setCursor(13,1);convertl1(cntl);delay(900);}
   lcd.setCursor(13,1);lcd.print("    ");


   lcd.setCursor(13,1);lcd.print("2   ");
     things_send();
     Serial.write("field4=");
     converts(amount);
     things_done();
     lcd.setCursor(13,1);lcd.print("    ");

   for(cntl=0;cntl<35;cntl++)
      {lcd.setCursor(13,1);convertl1(cntl);delay(900);}
   lcd.setCursor(13,1);lcd.print("    ");
 }
```

CHAPTER 7

CONCLUSION

Energy monitoring through the internet is easy. It gives the real power consumption as well as accurate reading. Also, it requires fewer labors and less time to monitor the energy. It can transmit the data to the utilities and also can receive information from utilities. After two months electricity bill will be paid otherwise supply line will be disconnected through the internet. After two months validity for alert purpose buzzer will be ON. It is easy to know the two months validity. By making this thing the energy will be monitored. The future scope will be on PC side one server software is required for automatic data collection. In this project, HTML knowledge is taken for demonstration purpose.

Future Scope:

This project can be extended using a GPS and GSM technology. GPRS is used to show details of the passport in weblinks

References

[1] Sanket Thakare, Akshay Shriyan, Vikas Thale, Prakash Yasarp, Keerthi Unni. "Implementation of an Energy Monitoring and Control Device based on IoT", 2016 IEEE [2] Altaf Hamed Shajahan, A.Anand. "Data acquisition and control using Arduino-Android Platform: Smart plug", 2013 IEEE [3] Wenjin Li, Xiaoqi Tan, Danny H.K. Tsang. "Smart Home Energy Management Systems Based on Non-Intrusive Load Monitoring", 2015 IEEE [4] H G Rodney tan, C H Lee and M H Mok. "Automatic power meter reading system using GSM network", 2007 IEEE [5] Sean Dieter Tebje Kelly, Nagender Kumar Suryadevara, and Subhas Chandra Mukhopadhyay "Towards the Implementation of IoT for Environmental Condition Monitoring in Home"2013 IEEE [6] Alessio Filippi, Ashish Pandharipande, Armand Lelkens, Ronald Rietman, "Multi-appliance power disaggregation: An approach to energy monitoring", 2010 IEEE International Energy Conference, 2010 IEEE [7] Laurent Lefˋevre, Olivier Mornard, Jean-Patrick Gelas, Maxime Morel "Monitoring Energy Consumption in Clouds: the Compatible One experience" 9th IEEE International Conference on Dependable, Autonomic and Secure Computing, 2011 IEEE [8] Paolo Barsocchi, Erina Ferro, Luigi Fortunati, Fabio Mavilia and Filippo Palumbo "Automatic power reading using GSM network" 2014 IEEE [9] Yasin Kabir, Yusuf Mohammad Mohsin1, and Mohammad Monirujaman Khan, "Automated Power Factor Correction and Energy Monitoring System" 2017 IEEE